APPENDIX E - TRANSACTION TESTS

1. Transaction Test Case Application Notes

Test Case Overview

Each test case follows the same format starting with a header page, the steps to perform the test, IRM testing, if applicable, and invalid testing, if applicable. The header provides the tester with a testing purpose, any prerequisites necessary to run the test, server connectivity requirements, and testing notes.

The test steps use three different types of external files to help perform the test; a transaction flat file, a Structured Query Language (SQL) script file, and a data sheet file. These three files work together as a set. One set of external files is used in the test case for each transaction processed. The flat file is used to generate external transactions at the XTP interface point. The SQL file is used to query the database to ensure that the transaction processed data properly and that the data elements were loaded properly. The data sheet files combine the information that will be contained in the results of the SQL script.

External Files Naming Convention

The naming convention used for the three external files used in the test cases is based on the transaction link names. Each transaction has a six-character link name for ease of identification. For example, the link name for the transaction TRANS-BQ-DATA-RECORD is DATEBT. The link name of the transaction(s) being tested appears in the test case purpose on the header page of each test case. The three types of files that use the link name are flat files, SQL script files, and data sheet files. The following is an example of an external file:

INITFF01.001

The first four letters of the file name are the first four letters of the link name. In the example above, the link name is INITHT. The next two characters are a code that designates one of the three external files. The three codes are as follows:

FF flat file
SS SQL script file
DS data sheet file

So the example file name above is a flat file due to the FF in the fifth and sixth position. The last two positions are filled with digits representing the sequence number of the test case. The example file name above can then be identified as a file used in test case 41422-1. The three digit extension is the sequence number of that particular transaction. One test case may have several occurrences of the same type of transaction. The example shows the extension .001, so this is the first INITHT transaction being used in test case 41422-1. If there were three different INITHT transactions being used in the same test case, then the last three digits of the file names would increase by one each time: INITFF01.002 and INITFF01.003.

Transaction Flat Files

A transaction flat file is one continuous string of ASCII characters, up to 1,273 long. No hard carriage returns should be present in the string, except at the very end of the file. Each of the 24 different transactions/linknames has its own format for the transaction string and can be found in the S&M User's Manual (UM). The UM lists each transaction in a table format and also breaks the string down into many different fields associated with that transaction. Each position of the string is important. The table explains each field, including its length, valid data, if it is required or not, and the type of data expected, such as A for alpha, N for numeric, or A/N for alphanumeric.

1. Creating Flat Files

To create a transaction flat file, use a text editor and type data for each of the fields listed in the UM tables, one right after the other. There should be no spaces between fields. Each character of each field listed in the table has to contain data (even if it is a blank space) for the length specified or errors will occur. Accuracy with the length of each field is very important for successful transactions. When the transaction string is finished, save it using the file naming conventions (discussed in the previous paragraph). If WordPerfect was used as the editor, remember to save the file as a DOS text file. Vi or Textedit can be used in a UNIX environment. If the flat files have been created on a PC, transfer them to the test server using File Transfer Protocol (FTP) procedures. Once the file is on the test server, edit out all of the carriage returns and other end of line characters that may exist, even if you can not see them. If the transaction file size is too large, the Vi editor will not edit the flat file due to string length limitations (the file is all one string). In the UNIX environment, the editor Textedit will accept larger string lengths and thus edit the file.

2. Capturing Flat Files

When a transaction is sent from the another server, a transaction flat file is automatically created and placed in a default directory. The system checks this default directory every two seconds and sends any file that is present to the next appropriate process according to the data in its header. To capture a "real" transaction flat file, the system needs to be delayed long enough for someone to make copies of the files in the default directory before they are sent on by the system. Two seconds is not long enough to copy files out of a directory, so the time defaults were changed from 2 seconds to 300 seconds (or five minutes), using the Vi editor. The file is called .SS_defaults and is located in the /h/SM/app-defaults directory if it ever needs to be changed again. The following shows what was changed in the .SS_defaults file:

```
tdc.MinimumSleepInterval        300
tdc.MaximumSleepInterval        300

tds.MinimumSleepInterval        300
tds.MaximumSleepInterval        300

tds_ds.MinimumSleepInterval     300
tds_ds.MaximumSleepInterval     300

*.tds_ready_dir default         /h/SM/data/tds/tds_ready
```

Once transactions have been sent from the main frame, the default directory should be monitored for arriving files. From the /h/SM/app-defaults directory, type "ls -l" to list the contents of the directory. It may take anywhere from a few seconds to several minutes for the transactions to arrive. The transaction flat file is identified by an extension of .cs1. Copy all files with the .cs1 extension to another directory (remember that you only have five minutes). By making a copy of the file instead of moving it out of the directory, it is possible to see if the transaction was successful or not. If you move the files to another directory they will not be processed. The file names are mostly numeric and very long. The naming convention used for these files is described in the *S&M Software Maintenance Manual (SMM), Appendix A*. There may be more than one transaction in a given file and it is recommended that an editor without line limits be used to edit these files. On SUN workstations this editor is called Textedit. The names are too long to use with FTP and should therefore be renamed first.

SQL Script Files

The SQL script files contain executable Oracle SQL statements that query the database for data used in the test cases. Each query was written to list the data pertaining to the transactions as a way to prove that the transaction did what was expected. The naming convention of the script files is the transaction link name with an extension of .sql. The scripts include automatic spooling of the SQL session and formatting for ease of reading. The scripts are executed from the directory where they reside using the 'start' command in SQL: 'start' <scriptname.sql>. The user will then be prompted for primary key values required to complete the query. These scripts are on file at SRA.

Data Sheet Files

Data Sheet Files have been generated to aid in determining if the results from the SQL scripts are correct. The data sheets are a list of the transaction fields and the values that should be in the field. The data sheets were generated using WordPerfect macros that parse the flat files as one long string and break it down into individual fields with their field name. The resulting data sheets matching these test cases are on file with the CM department at SRA.

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)	
Purpose: The purpose of this test is to verify that the software correctly processes TRANS-XXX-FORCE-MODULE (DESCDT, INDXDT, TITLDT, DLFMDT, JJDSDT) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 16 transaction flat files used with XTP c. 5 SQL script files d. 15 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. FMs P0Y and PBA already exist
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and IRM sources. Transactions DESCDT and JJDSDT have been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start titldt*** to run the TITLDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
5	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/titlff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A TITLDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the title data of Force Module P0Y for OPLAN 2591H.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A TITLDT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start titldt*** to run the TITLDT.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the TITLDS01.001 data sheet.		
9	From the SQL> prompt on all three servers, type ***start descdt*** to run the DESCDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
10	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/descff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DESCDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add 10 lines of description to Force Module P0Y for OPLAN 2591H.		
11	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DESCDT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start descdt*** to run the DESCDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DESCDS01.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
13	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/descff01.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DESCDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the first line of description to Force Module P0Y for OPLAN 2591H.		
14	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DESCDT transaction should be present on the receive queues of both servers.		
15	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start descdt*** to run the DESCDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DESCDS01.002 data sheet. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
16	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/descff01.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DESCDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will try to add a force module description to an OPLAN (9999K) that does not exist. An error will occur stating the OPLAN does not exist.		
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start jjdsdt1*** to run the JJDSDT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/jjdsff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A JJDSDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update (change) the title data of FM PBA for OPLAN 2591H.		
19	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A JJDSDT transaction should be present on the receive queues of both servers.		
20	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start jjdsdt1*** to run the JJDSDT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the JJDSDS01.001 data sheet.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
21	From the SQL> prompt on all three servers, type ***start jjdsdt2*** to run the DESCDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/jjdsff01.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A JJDSDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the description data of FM PBA for OPLAN 2591H.		
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A JJDSDT transaction should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start jjdsdt2*** to run the JJDSDT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the JJDSDS01.002 data sheet.		
25	From the SQL> prompt on all three servers, type ***start jjdsdt3_4*** to run the JJDSDT3_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
26	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/jjdsff01.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A JJDSDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add ULNs, PAAC, PAAD, PAAF, and PAAK to FM PBA of OPLAN 2591H.		
27	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A JJDSDT transaction should be present on the receive queues of both servers.		
28	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start jjdsdt3_4*** to run the JJDSDT3_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the JJDSDS01.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
29	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/jjdsff01.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A JJDSDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add CINs, AR20026, AR20027, and AR20028 to FM PBA of OPLAN 2591H.		
30	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A JJDSDT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
31	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start jjdsdt3_4*** to run the JJDSDT3_4.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the JJDSDS01.004 data sheet.		
32	From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
33	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add ULN PAAJ to FM PBA for OPLAN 2591H.		
34	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		
35	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
36	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add CIN AR12345 to FM PBA for OPLAN 2591H.		
37	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		
38	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
39	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add PIN NM12345 to FM PBA for OPLAN 2591H.		
40	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
41	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
42	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete ULN PAAJ from FM PBA for OPLAN 2591H.		
43	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		
44	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.004 data sheet. ULN PAAJ should not be present. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
45	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queue. This transaction will delete CIN AR12345 from FM PBA for OPLAN 2591H.		
46	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		
47	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.001 data sheet. CIN AR12345 should not be present. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
48	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/indxff01.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INDXDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete PIN NM12345 to FM PBA for OPLAN 2591H.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
49	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INDXDT transaction should be present on the receive queues of both servers.		
50	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start indxdt*** to run the INDXDT.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INDXDS01.006 data sheet. PIN NM12345 should not be present. Remember this data as a check for the next transaction.		
51	From the SQL> prompt on all three servers, type ***start dlfmdt*** to run the DLFMDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
52	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dlfmff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DLFMDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete FM PBA from OPLAN 2591H.		
53	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An DLFMDT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-1: TRANS-XXX-FORCE-MODULE (DESCDT-INDXDT-TITLDT-DLFMDT-JJDSDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
54	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start dlfmdt*** to run the DLFMDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DLFMDS01.001 data sheet. This will exit SQL*Plus and start up System Services.		
55	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/undoff01.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DLFMDT and two JJDSDT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will undo what all of the previous transactions did for OPLAN 2591H.		
56	Verify that all of the previous INITHT, TITLDT, DESCDT, JJDSDT, and DLFMDT transactions generated were received at the temporary directory created for XTDS for all three servers	All five different types of transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
57	Using the Audit Reports, obtain a printout of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also, each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-2: TRANS-FM-EXCLUDE (STRYDT)	
Purpose: The purpose of this test is to verify that the software correctly processes TRANS-FM-EXCLUDE (STRYDT) transaction.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 1 transaction flat file used with XTP c. 1 SQL script file d. 1 result data sheet e. OPLAN 2591H exists and is a copy of 2500T f. FM RRR does not already exist
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant server by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-2: TRANS-FM-EXCLUDE (STRYDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e. xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	Enter the following: Transaction File Name:***/h/SM/data/tds/krh/add_fm02.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A TITLDT, DESCDT, and three INDXDT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add FM RRR and all its requirements to OPLAN 2591H, setting up for the STRYDT test . **Note:** If this transaction fails due to a duplicate tuple error, it means that the FM already exists. Disregard the failure and continue on with the test. The FM is primarily to set up for the rest of the test.		
6	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A TITLDT, DESCDT, and three INDXDT transactions should be present on the receive queues of both the Destination and Distant servers.		

TEST CASE 41422-2: TRANS-FM-EXCLUDE (STRYDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start strydt*** to run the STRYDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
8	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/stryff02.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A STRYDT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete FM RRR and all its requirements from OPLAN 2591H.		
9	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A STRYDT transaction should be present on the receive queues of both the Destination and Distant servers.		
10	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start strydt*** to run the STRYDT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the STRYDS02.001 data sheet. FM RRR should not be there. This will exit SQL*Plus and start up System Services.		
11	Verify that all of the previous STRYDT transactions generated were received at the temporary directory created for XTDS for all three servers	The STRYDT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		

TEST CASE 41422-2: TRANS-FM-EXCLUDE (STRYDT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Using the Audit Reports, obtain a printout of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)	
Purpose: The purpose of this test is to verify that the software correctly processes the TRANS-BQ-DATA-RECORD (DATEBT) transaction which is used to change RLD, ALD, EAD, LAD, and RDD dates for requirements in an OPLAN.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 15 transaction flat files used with XTP c. 1 SQL script files d. 15 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. FMID PXX exists for OPLAN 2591H g. ULNs PA3-PA3F exist on OPLAN 2591H h. CINs AR20000-AR20003 exist on OPLAN 2591H I. PINs NM21001-NM21004 exist on OPLAN 2591H
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	If the FMID PXX or any of the ULNs, CINs, or PINS do not exist, start up the System Services application and enter the following: Transaction File Name: ***/h/SM/data/tds/krh/fix_date*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	One JJDSDT, seven ULNUBTs, eight NRNUBTs, and three INDSDTs transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create the FMID, ULNs, CINS, and PINs if they are missing. If they already exist, the transactions will fail as duplicate tuples in the database.		
6	Using SQL*Plus, run the RESET_ALL.SQL script on the Source, Destination, and Distant servers, and start up the System Services application server	This will reset all five of the date values (ALD, LAD, EAD, RLD, and RDD) for the ULNs, CINs, and PINs used in this test case.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt1_4*** to run the DATEBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
8	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the LAD and EAD dates up by 2, excluding the RLD and RDD dates for the appropriate ULNs. The OPLAN is 2591H, the base date is ALD, and the ULN range is from PA3A to PA3F.		
9	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
10	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt1_4*** to run the DATEBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
11	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD, and RDD dates down by 3 for the appropriate ULNs. The OPLAN is 2591H, the base date is ALD, and the FMID is PXX.		
12	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
13	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt1_4*** to run the DATEBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the RLD, EAD, LAD, and RDD dates based on the Air to CONUS mode table for the appropriate ULNs. The OPLAN is 2591H, the base date is ALD, and the range is from PA3A to PA3F.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt1_4*** to run the DATEBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
17	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD and RDD dates up by 3 (ignore mode table) for the appropriate ULNs. The OPLAN is 2591H, the base date is ALD, and the range is from PA3A to PA3F.		
18	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
19	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt1_4*** to run the DATEBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.004 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
20	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will test for an invalid FM ID for ULNs. An error message will appear and be recorded in the **msg** file.		
21	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt6_9*** to run the DATEBT6_9.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the LAD and EAD dates up by 2, excluding the RLD and RDD dates for the appropriate CINs. The OPLAN is 2591H, the base date is ALD, and the CIN range is from AR20000 to AR20003.		
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt6_9*** to run the DATEBT6_9.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.006 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
25	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.007*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD, and RDD dates down by 3 for the appropriate CINs. The OPLAN is 2591H, the base date is ALD, and the FMID is PXX.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
26	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
27	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt6_9*** to run the DATEBT6_9.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.007 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
28	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.008*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the RLD, EAD, LAD, and RDD dates based on the Air to CONUS mode table for the appropriate CINs. The OPLAN is 2591H, the base date is ALD, and the CIN range is from AR20000 to AR20003.		
29	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
30	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt6_9*** to run the DATEBT6_9.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.008 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
31	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.009*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD and RDD dates up by 3 (ignore mode table) for the appropriate CINs. The OPLAN 2591H, the base date is ALD, and the CIN range is from AR20000 to AR20003.		
32	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
33	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt6_9*** to run the DATEBT6_9.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.009 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
34	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.010*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will test for an invalid FM ID for CINs. An error message will appear and be recorded in the **msg** file.		
35	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt11_14*** to run the DATEBT11_14.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
36	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.011*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the LAD and EAD dates up by 2, excluding the RLD and RDD dates for the appropriate PINs. The OPLAN is 2591H, the base date is ALD, and the PIN range is from NM21000 to NM21004.		
37	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
38	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt11_14*** to run the DATEBT11_14.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.0011 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
39	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.012*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD, and RDD dates down by 4 for the appropriate PINs. The OPLAN is 2591H, the base date is ALD, and the FMID is PXX.		
40	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
41	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt11_14*** to run the DATEBT11_14.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.0012 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
42	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.013*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the RLD, EAD, LAD, and RDD dates based on the Air to CONUS mode table for the appropriate PINs. The OPLAN is 2591H, the base date is ALD, and the PIN range from NM21001 to NM21004.		
43	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
44	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt11_14*** to run the DATEBT11_14.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.0013 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
45	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.014*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will adjust the RLD, EAD, LAD and RDD dates up by 3 (ignore mode table) for the appropriate PINs. The OPLAN 2591H, the base date is ALD, and the PIN range is from NM21001 to NM21004.		
46	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DATEBT transaction should be present on the receive queues of both servers.		
47	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start datebt11_14*** to run the DATEBT11_14.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the DATEDS03.0014 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
48	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/dateff03.015*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DATEBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will test for an invalid FM ID for PINs. An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-3: TRANS-BQ-DATA-RECORD (DATEBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
49	Verify that all of the previous DATEBT transactions generated were received at the temporary directory created for XTDS for all three servers	The DATEBT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
50	Using the Audit Reports, obtain a printout of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-4: TRANS-JD-FORCE-DATA (ULNUBT)	
Purpose: The purpose of this test is to verify that the software correctly processes the TRANS-JD-FORCE-DATA (ULNUBT) transaction which is used to add, change and delete requirements to/from an OPLAN.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 3 transaction flat files used with XTP c. 1 SQL script files d. 3 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. ULN PDXC4 does not already exist on OPLAN 2591H
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-4: TRANS-JD-FORCE-DATA (ULNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start ulnubt*** to run the ULNUBT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/ulnuff04.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A ULNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add ULN PDXC4 to OPLAN 2591H.		

TEST CASE 41422-4: TRANS-JD-FORCE-DATA (ULNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A ULNUBT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start ulnubt*** to run the ULNUBT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the ULNUDS04.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
9	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/ulnuff04.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A ULNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the force description field and the authorized personnel field for ULN PDXC4 for OPLAN 2591H.		
10	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A ULNUBT transaction should be present on the receive queues of both servers.		
11	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start ulnubt*** to run the ULNUBT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the ULNUDS04.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-4: TRANS-JD-FORCE-DATA (ULNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/ulnuff04.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A ULNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete ULN PDXC4 from OPLAN 2591H.		
13	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A ULNUBT transaction should be present on the receive queues of both servers.		
14	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start ulnubt*** to run the ULNUBT .SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the ULNUDS04.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
15	Verify that all of the previous ULNUBT transactions generated were received at the temporary directory created for XTDS for all three servers	The ULNUBT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
16	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)	
Purpose: The purpose of this test is to verify that the software correctly processes TRANS-NON-UNIT-Data (NRNUBT) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 28 transaction flat files used with XTP c. 1 SQL script files d. 28 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. PIN NM21005 and CIN NS21005 don't already exist on OPLAN 2591H
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers; a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt1_4*** to run the NRNUBT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add non-unit personnel data, PIN NM21005, to OPLAN 2591H. The RDD is set to 118, the ALD is set to 111, the LAD is set to 118, and the EAD is set to 113.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt1_4*** to run the NRNUBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
9	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the ALD field from 111 to 222 in the non-unit personnel data on OPLAN 2591H.		
10	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
11	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt1_4*** to run the NRNUBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete field POE ALD from the non-unit personnel data of OPLAN 2591H, leaving it blank.		
13	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
14	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt1_4*** to run the NRNUBT1_4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
15	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete all the non-unit personnel data, PIN NM21005, from OPLAN 2591H.		
16	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt1_4*** to run the NRNUBT1_4.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.004 data sheet.		
18	From the SQL> prompt on all three servers, type ***start nrnubt5_8*** to run the NRNUBT5_8.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
19	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add non-unit cargo data, CIN NS21005, to OPLAN 2591H. The RDD is set to 118, the ALD is set to 111, the LAD is set to 118, and the EAD is set to 113.		
20	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
21	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt5_8*** to run the NRNUBT5_8.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.005 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the cargo STONS field from 300 to 500 and update the cargo description field of the non-unit cargo data of OPLAN 2591H.		
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt5_8*** to run the NRNUBT5_8.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.006 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
25	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.007*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete the value in the cargo STONS field, leaving it zero, from the non-unit cargo data of OPLAN 2591H.		
26	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
27	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt5_8*** to run the NRNUBT5_8.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.007 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
28	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.008*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete non-unit cargo data, CIN NS21005, from OPLAN 2591H.		
29	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NRNUBT transaction should be present on the receive queues of both servers.		
30	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nrnubt5_8*** to run the NRNUBT5_8.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NRNUDS05.008 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
31	Verify that all of the previous NRNUBT transactions generated were received at the temporary directory created for XTDS for all three servers	The NRNUBT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
32	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps. Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		
INVALID FIELD TEST				
33	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.009*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid OPLAN. An error message will appear and be recorded in the **msg** file.		
34	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.010*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an OPLAN (a key field) for null. An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
35	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.011*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test a Movement Requirement ID (a key field) for null. An error message will appear and be recorded in the **msg** file concerning invalid data.		
36	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.012*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test a Cargo Type Code (a key field) for null. An error message will appear and be recorded in the **msg** file.		
37	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.013*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test a Cargo Extension Code (a key field) for null. An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
38	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.014*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test a Cargo Container Code (a key field) for null. An error message will appear and be recorded in the **msg** file.		
39	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.015*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Record Type (not personnel or cargo). An error message will appear and be recorded in the **msg** file concerning invalid data.		
40	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.016*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Origin Geographic Location (one that does not exist). An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
41	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.017*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid POE Geoloc (one that does not exist). An error message will appear and be recorded in the **msg** file.		
42	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.018*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid POE ALD (one that does not exist). An error message will appear and be recorded in the **msg** file.		
43	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.019*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid POE EDD (one that does not exist). An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
44	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.020*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid POD EAD (one that does not exist). An error message will appear and be recorded in the **msg** file.		
45	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.021*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid POD Preferred Mode (Not A, L, S, P, X, or Z). An error message will appear and be recorded in the **msg** file.		
46	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.022*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Destination Preferred Mode (Not A, L, S, P, X, or Z). An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
47	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.023*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Persons Requiring Transportation (non-numeric). An error message will appear and be recorded in the **msg** file.		
48	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.024*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Cargo Type Code (one that does not exist, not A, B, C, D, E, F, G, H, J, K, L, M, N, P, or R). An error message will appear and be recorded in the **msg** file.		
49	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.025*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Cargo Container Code (one that does not exist, Not A, B, C, or D). An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-5: TRANS-NON-UNIT-DATA (NRNUBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
50	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.026*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Cargo STONS (non-numeric). An error message will appear and be recorded in the **msg** file.		
51	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.027*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will test an invalid Create Date. An error message will appear and be recorded in the **msg** file.		
52	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nrnuff05.028*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NRNUBT transaction will not be generated. This transaction will attempt to update a Cargo Record which does not exist. An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)	
Purpose: The purpose of this test is to verify that the software correctly processes the TRANS-NS-DATA (NSCGBT) transaction which is used to add and delete tonnage requirements to/from an OPLAN.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 7 transaction flat files used with XTP c. 2 SQL script files d. 7 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. ULN PNSC does not already exist on OPLAN 2591H
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start ulnubt2*** to run the ULNUBT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	If ULN PNSC already exists under OPLAN 2591H, in the previous step, then skip the following two steps If ULN PNSC does not exist, the next two steps will create it	ULN PNSC is required to continue this test case.		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/ulnuff06.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A ULNUBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create ULN PNSC for OPLAN 2591H.		
8	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A ULNUBT transaction should be present on the receive queues of both servers.		
9	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
10	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nscgff06.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NSCGBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add Cargo Category Codes (CATs) G3D and L2D to ULN PNSC for OPLAN 2591H (third level detail).		
11	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NSCGBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the NSCGDS06.001 data sheet.		
13	From the SQL> prompt on all three servers, type ***start nscgbt2*** to run the NSCGBT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nscgff06.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NSCGBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add cargo detail data to CAT C1D of ULN PNSC for OPLAN 2591H (fourth level detail).		
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NSCGBT transaction should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt2*** to run the NSCGBT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the NSCGDS06.002 data sheet.		
17	From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nscgff06.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NSCGBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will change data in CAT G3D of ULN PNSC for OPLAN 2591H (third level detail).		
19	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NSCGBT transaction should be present on the receive queues of both servers.		
20	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the NSCGDS06.003 data sheet.		
21	From the SQL> prompt on all three servers, type ***start nscgbt2*** to run the NSCGBT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nscgff06.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NSCGBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete data in CAT G3D of ULN PNSC for OPLAN 2591H (fourth level detail).		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NSCGBT transaction should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt2*** to run the NSCGBT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the NSCGDS06.004 data sheet.		
25	From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
26	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/nscgff06.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An NSCGBT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete the data in CAT G3D and L2D of ULN PNSC for OPLAN 2591H (third level detail).		
27	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An NSCGBT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-6: TRANS-NS-DATA (NSCGBT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
28	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start nscgbt1*** to run the NSCGBT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the NSCGDS06.005 data sheet. This will exit SQL*Plus and start up System Services.		
29	Verify that all of the previous NSCGBT transactions generated were received at the temporary directory created for XTDS for all three servers	The NSCGBT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
30	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)	
Purpose: The purpose of this test case is to verify that the software properly processes TRANS-SM-CARRIER-DATA (SCHDET) and TRANS-SM-CARRIER-NEW-DATA (SCHPET) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 5 transaction flat files used with XTP c. 2 SQL script files d. 5 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and S&M Sources.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff07.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add carrier SFCA500 to OPLAN 2591H.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.001 and SCHPDS07.001 data sheets.		
9	From the SQL> prompt on all three servers, type ***start schdet2*** to run the SCHDET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
10	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff07.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add an itinerary to carrier SFCA500, with geolocs of RXLD and UBEA, stop codes of O and U.		
11	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
12	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet2*** to run the SCHDET2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.002 and SCHPDS07.002 data sheets.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
13	From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff07.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update carrier SFCA500, changing the Carrier Type from TEST to TESTING.		
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET transaction should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.003 data sheet.		
17	From the SQL> prompt on all three servers, type ***start schdet2*** to run the SCHDET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff07.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete itinerary leg RXLD from carrier SFCA500.		
19	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET transaction should be present on the receive queues of both servers.		
20	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet2*** to run the SCHDET2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.004 data sheet.		
21	From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff07.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete carrier SFCA500 from OPLAN 2591H and the database.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET transaction should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet1*** to run the SCHDET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.005 data sheet. This will exit SQL*Plus and start up System Services.		
25	Verify that all of the previous SCHDET and SCHPET transactions generated were received at the temporary directory created for XTDS for all three servers	All SCHDET and SCHPET transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
The following test verifies that the S&M Applications program still generates the SCHDET and SCHPET transactions.				
26	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet5*** to run the SCHDET5.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
27	On the Source server, go to the following: Click on **Scheduling and Movement ▸ Add New Carrier and Itinerary ▸ Air Non-Cargo/Non-PAX Carrier**	The **Add Air Cargo/PAX Carrier** screen appears (SM-A01-A).		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
28	Enter and transmit the following data: Carrier ID: ***SFCA501*** Configuration: ***Test*** Carrier Type: ***Test*** Click on **Source: AMC** Comment: ***S&M Test*** Supported OPLAN: ***2591H*** Loc: ***VBEA*** Stp: ***O*** Depart: ***121200ZJUN95*** Loc: ***SBEA*** Stp: ***U*** Arrive: ***131200ZJUN95***	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add carrier SFCA501 to OPLAN 2591H.		
29	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
30	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet5*** to run the SCHDET5.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.006 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
31	From the Scheduling and Movement screen (SM-000-1) enter **dele** at the command line and hit **Transmit**	The **Delete Carrier** screen (SM-D01-1) will appear.		
32	Enter **SFCA501** at the first Carrier ID location and hit **Transmit** Enter **C** for confirmation	Carrier SFCA501 will be deleted.		

TEST CASE 41422-7: TRANS-SM-CARRIER-DATA/TRANS-SM-CARRIER-NEW-DATA (SCHDET-SCHPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
33	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET transaction should be present on the receive queues of both servers.		
34	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet5*** to run the SCHDET5.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS07.007 data sheet. This will exit SQL*Plus and start up System Services.		
35	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)	
Purpose: The purpose of this test is to verify that the software properly processes TRANS-SM-MANIFEST-DATA (MANIET) and TRANS-SM-MANIFEST-NEW-DATA (MANPET) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 10 transaction flat files used with XTP c. 2 SQL script files d. 10 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and S&M Sources.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet3*** to run the SCHDET3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	If carrier KKK123 already exists under OPLAN 2591H in the previous step, then skip the following two steps If carrier KKK123 does not exist, the next two steps will create it	Carrier KKK123 is required to continue this test case.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff08.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add carrier KKK123 to OPLAN 2591H for the purpose of the following MANIET and MANPET transactions. The geolocs SBEA, VBEA, DKSD are onloads and HNTS is the offload. PAX will be 200, MTONS will be 100,000, SQFT will be 75,000, and MBBLS will be 400.		
8	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
9	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet3*** to run the SCHDET3.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS08.001 and SCHPDS08.001 data sheets.		
10	From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
11	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will add ULN PAZGC to carrier KKK123. The PAX is 40, the MTONS is 0, the SQFT is 40, and the MBBLS is 0.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
13	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will change the allocation by adding ULN PAZSC to carrier KKK123. PAX is 40, MTONS is 0, SQFT is 29148, and MBBLS is 0.		
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.001 data sheet.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
17	From the SQL> prompt on all three servers, type ***start maniet2*** to run the MANIET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will manifest ULNs PAZGC and PAZSC to carrier KKK123 of OPLAN 2591H. PAX will be 0 for both, MTONS will be 0 for both, SQFT will be 27515 and 29108 respectively, and MBBLS will be 0 for both.		
19	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
20	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet2*** to run the MANIET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
21	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will change the manifest by adding 40 to the SQFT of both PAZGC and PAZSC of carrier KKK123 of OPLAN 2591H. SQFT will be 27556 and 29148, respectively.		
22	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
23	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet2*** to run the MANIET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.004 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
24	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will de-manifest ULNs PAZGC and PAZSC from carrier KKK123 of OPLAN 2591H.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
25	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET transaction should be present on the receive queues of both servers.		
26	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet2*** to run the MANIET2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.005 data sheet.		
27	From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
28	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/maniff08.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A MANIET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will de-allocate ULNs PAZGC and PAZSC from carrier KKK123 of OPLAN 2591H.		
29	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET transaction should be present on the receive queues of both servers.		
30	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the MANIDS08.006 data sheet.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
31	From the SQL> prompt on all three servers, type ***start schdet3*** to run the SCHDET3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
32	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/schdff08.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete carrier KKK123 from OPLAN 2591H.		
33	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET transaction should be present on the receive queues of both servers.		
34	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet3*** to run the SCHDET3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS08.002 data sheet. This will exit SQL*Plus and start up System Services.		
35	Verify that all of the previous MANIET and MANPET transactions generated were received at the temporary directory created for XTDS for all three servers	All MANIET and MANPET transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
36	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
The following test verifies that the S&M Applications program still generates the MANIET and MANPET transactions.				
37	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
38	On the Source server, from the **Scheduling and Movement** menu (SM-000-1), go to the following: Click on **Scheduling and Movement ▸ Add New Carrier and Itinerary ▸ Sea Cargo/PAX Carrier**	The **Add Sea Cargo/PAX Carrier** screen appears (SM-A01-S).		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
39	Enter and transmit the following: Carrier ID: ***KKK123*** Ship Name: ***USS Minnow*** IRCS: ***M_Test*** Carrier Type: ***M_Test*** Comment: ***USS Minnow Test*** Click on: **Source: MTMC** Supported OPLAN: ***2591H*** PAX: ***200*** MTONS: ***100000*** SQFT: ***75000*** MBBLS: ***400*** Act Loc Stp Arrive Depart ***SBEA O 121010ZJUN95*** ***VBEA O 131010ZJUN95 141010ZJUN95*** ***DKSD O 151010ZJUN95 161010ZJUN95*** ***HNTS U 171010ZJUN95***	Data is accepted. A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues.		
40	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS08.001 and SCHPDS08.001 data sheets.		
41	From the SQL> prompt on all three servers, type ***start maniet1*** to run the MANIET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
42	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Add Group Allocations**	The **Add Group Allocations** screen appears (SM-PO2-1).		
43	Enter and transmit the following: Carrier ID: ***KKK123*** Onload: ***SBEA*** Offload: ***HNTS*** OPLAN: ***2591H*** Click on **Over Allocated** Click on **Fully Allocated** Click on **Part Allocated** Click on **Not Allocated**	The **Add Group Requirement Allocations - Sea** screen appears (SM-PO2-S). ULNs PAZSC and PAZGC should appear in the bottom left corner of the screen.		
44	Enter and transmit the following: Click on **Accept button for PAZSC** PAX: ***10*** MTONS: ***10*** SQFT: ***10*** MBBLS: ***10*** Click on **Accept button for PAZGC** PAX: ***10*** MTONS: ***10*** SQFT: ***10*** MBBLS: ***10***	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will allocate ULNs PAZSC and PAZGC to carrier KKK123. **Note:** Follow through with the requirement overload.		
45	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
46	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Add/Review/Modify Allocation**	The **Add, Review, or Modify Allocations** screen appears (SM-PO3-1).		
47	Enter and transmit the following: Carrier ID: ***KKK123*** OPLAN: ***2591H*** Click on **Data to Review/Modify: All Allocations**	The **Add, Rev, or Mod Allocated Rqmts by ONLD/OFLD Sea** screen appears (SM-PO3-S).		
48	Enter a ***d*** in the Act column next to ULN PAZSC and **Transmit** **Note:** Follow through with the requirement overload	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will change the allocation by deleting ULN PAZSC from carrier KKK123.		
49	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
50	Using SQL*Plus, run the MANIET.SQL and MANPET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the MANIDS08.002 data sheet.		
51	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Add Group Manifests**	The **Add Group Manifests** screen appears (SM-MO2-1).		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
52	Enter and transmit the following: Carrier ID: ***KKK123*** Onload: ***SBEA*** Offload: ***HNTS*** OPLAN: ***2591H*** Click on **Over Allocated** Click on **Fully Allocated** Click on **Part Allocated** Click on **Not Allocated**	The **ADD GROUP REQUIREMENT MANIFESTS - SEA** screen appears (SM-MO2-S). ULNs PAZSC and PAZGC should appear in the bottom left corner of the screen.		
53	Enter and transmit the following: Click on **Accept Button for PAZSC** Click on **Accept Button for PAZGC**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will manifest ULNs PAZSC and PAZGC to carrier KKK123.		
54	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
55	Using SQL*Plus, run the MANIET.SQL and MANPET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the MANIDS08.003 data sheet.		
56	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Add/Review/Modify Manifests**	The **Add, Review, or Modify Manifests** screen appears (SM-MO3-1).		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
57	Enter and transmit the following: Carrier ID: ***KKK123*** OPLAN: ***2591H*** Click on **Data to Review/Modify: All Allocations**	The **Add, Rev, or Mod, Manifested Rqmts by ONLD/OFLD Sea** screen appears (SM-MO3-S).		
58	Enter a ***d*** in the Act column next to ULN PAZSC and **Transmit**	A MANIET and MANPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will change the manifest by deleting ULN PAZSC from carrier KKK123.		
59	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET and MANPET transaction should be present on the receive queues of both servers.		
60	Using SQL*Plus, run the MANIET.SQL and MANPET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the MANIDS08.004 data sheet.		
61	From the Scheduling and Movement menu: Click on **Allocations/Manifests** Click on **Demanifest Carrier**	The **Demanifest Carrier** screen appears (SM-MO5-1).		
62	Enter and transmit the following: Click on **Demanifest Carrier ID** Demanifest Carrier ID: ***KKK123***	The **Demanifest Carrier Confirmation** screen appears (SM-MO5-2).		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
63	Enter a ***C*** and **Transmit**	A MANIET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will de-manifest ULN PAZGC from carrier KKK123 of OPLAN 2591H.		
64	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET transaction should be present on the receive queues of both servers.		
65	Using SQL*Plus, run the MANIET.SQL and MANPET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the MANIDS08.005 data sheet.		
66	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Deallocate Carrier**	The **Deallocate Carrier** screen appears (SM-PO5-1).		
67	Enter and transmit the following: Click on **Deallocate Carrier ID** Deallocate Carrier ID: ***KKK123***	The **Deallocate Carrier Confirmation** screen appears (SM-PO5-2).		
68	Enter a ***C*** and **Transmit**	A MANIET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will de-allocate ULN PAZGC from carrier KKK123 of OPLAN 2591H.		

TEST CASE 41422-8: TRANS-SM-MANIFEST-DATA/TRANS-SM-MANIFEST-NEW-DATA (MANIET-MANPET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
69	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A MANIET transaction should be present on the receive queues of both servers.		
70	Using SQL*Plus, run the MANIET.SQL and MANPET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the MANIDS08.006 data sheet.		
71	From the Scheduling and Movement menu: Click on **Allocations/Manifests ▸ Delete Carrier**	The **Delete Carrier** screen appears (SM-DOM-1).		
72	Enter ***KKK123*** for the Carrier ID in box 1 and **Transmit**	The **Delete Carrier Warning** screen appears (SM-DOM-1.1).		
73	Enter a ***C*** and **Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will delete carrier KKK123 of OPLAN 2591H.		
74	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)

Purpose: The purpose of this test case is to verify that valid incoming TRANS-SM-DIV-CHANGE-DATA (DICHET) transactions are correctly processed by S&M/CS System Services software. Included in this processing is the correct channeling of exercise and real-world data, data validation checking and database update. This test case employs three sets of steps which exercise the three possible function codes for this transaction type. These are C (change), D (divert), and R (remarks).

Tester Information:	**Prerequisites for this test:**
Tester Name: Phone Number: Date(s) of Test:	a. 3 servers with the proper permissions established. b. 2 transaction flat files used with XTP c. 2 SQL script files d. 2 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and S&M Sources.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	If carrier SFCA555 already exists under OPLAN 2591H in the previous step, then skip the following two steps If carrier SFCA555 does not exist, the next two steps will create it	Carrier SFCA555 is required to continue this test case.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Enter the following: Transaction File Name:***/h/SM/data/tds/krh/schdff09.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions** **Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create carrier SFCA555 on OPLAN 2591H, setup for all the following DICHET's to use.		
8	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
9	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS09.001 and SCHPDS09.001 data sheets.		
10	From the SQL> prompt on all three servers, type ***start dichet1*** to run the DICHET1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
11	Enter the following: Transaction File Name:***/h/SM/data/tds/krh/dichff09.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions** **Transmit**	A DICHET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will change the arrival date from 13 June to 14 June for carrier SFCA555 of OPLAN 2591H.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DICHET transaction should be present on the receive queues of both servers.		
13	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start dichet1*** to run the DICHET1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the DICHDS09.001 data sheets.		
14	From the SQL> prompt on all three servers, type ***start dichet2*** to run the DICHET2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
15	Enter the following: Transaction File Name:***/h/SM/data/tds/krh/dichff09.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A DICHET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the remarks field for carrier SFCA555 of OPLAN 2591H.		
16	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DICHET transaction should be present on the receive queues of both servers.		
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start dichet2*** to run the DICHET2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the DICHDS09.002 data sheets.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
19	Enter the following: Transaction File Name:***/h/SM/data/tds/krh/schdff09.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SCHDET and SCHPET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will delete carrier SFCA555 from OPLAN 2591H.		
20	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SCHDET and SCHPET transaction should be present on the receive queues of both servers.		
21	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start schdet4*** to run the SCHDET4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SCHDDS09.002 data sheet. This will exit SQL*Plus and start up System Services.		
22	Verify that all of the previous DICHET transactions generated were received at the temporary directory created for XTDS for all three servers	The DICHET transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
The following test verifies that the S&M Applications program still generates the DICHET transactions.				
23	Repeat Steps 4 through 9 from above.	Carrier SFCA555 will be added to OPLAN 2591H.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
24	From the Scheduling and Movement menu: Click on **Review/Modify Carrier**	The **Specify Criteria for Carriers** screen appears (SM-P00-1).		
25	Enter ***SFCA555*** in the Carrier IDs to Review and press **Transmit** twice.	The **Review or Modify Carrier - Display Options** screen appears (SM-PA03-3).		
26	Enter and transmit the following: Click on **Carrier and Itinerary** Click on **Allocations: All Allocations** Click on **Manifests: All Manifests** OPLAN: ***2591H***	The **Review or Modify Carrier and Itinerary - Air** screen appears (SM-PA03-A).		
27	Change the Arrival date for SBEA from 131010ZJUN95 to 141010ZJUN95 and **Transmit**	A DICHET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the arrival date by one day for carrier SFCA555 of OPLAN 2591H.		
28	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DICHET transaction should be present on the receive queues of both servers.		
29	Enter a comment in the second remarks text field and **Transmit**	A DICHET transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the remarks field for carrier SFCA555 of OPLAN 2591H.		

TEST CASE 41422-9: TRANS-SM-DIV-CHG-DATA (DICHET)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
30	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A DICHET transaction should be present on the receive queues of both servers.		
31	Using SQL*Plus, run the DICHET.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the DICHDS09.001 and DICHFD09.002 data sheets.		
32	Using the Audit Reports, obtain a printout of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)	
Purpose: The purpose of this test case is to demonstrate that the software correctly processes all variations of the TRANS-CH-PLAN-INIT (INITHT) transaction.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 5 transaction flat files used with XTP c. 1 SQL scripts d. 5 result data sheets e. OPLAN 2591H exists and is a copy of 2500T f. USERIDs are loaded with OPLAN permissions
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and IRM sources. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Each time an OPLAN is initialized, deleted, or modified.

5. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login and bring up System Services on all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues This transaction will create a Normal, Local, and Unclassified OPLAN, number 2577K.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.001 data sheets.		
9	From the SQL> prompt on all three servers, type ***start initht2*** to run the INITHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
10	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create a Restricted, Local, and Unclassified OPLAN, number 3577K. The restriction will be for users SMTEST4, SMTEST1 and SMTEST2.		
11	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
12	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht2*** to run the INITHT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.002 data sheets.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
13	From the SQL> prompt on all three servers, type ***start initht3*** to run the INITHT3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add user SMTEST3 to the restricted user list of OPLAN 3577K.		
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht3*** to run the INITHT3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
17	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions** **Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete users SMTEST2 and SMTEST3 from the restricted user list of OPLAN 3577K.		
18	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
19	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht3*** to run the INITHT3.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.004 data sheets.		
20	From the SQL> prompt on all three servers, type ***start initht2*** to run the INITHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
21	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions** **Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will change OPLAN 3577K from Restricted to Normal.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
22	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
23	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht2*** to run the INITHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.005 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
24	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff10.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will set the network flag to 1 (from 0) which means OPLAN 3577K is distributed.		
25	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
26	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht2*** to run the INITHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INITDS10.006 data sheet. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
27	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
28	Enter and transmit the following data: OPLAN: ***2577K and 3577K*** Click on **Delete** **Note:** Delete the plans one at a time.	A RESTHT and two SYNCHT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will delete OPLAN 2577K and 3577K.		
29	Verify that all of the INITHT transactions generated were received at the temporary directory created for XTDS for all three servers	The INITHT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
IRM Transaction Test				
30	Login and bring up System Services on all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
31	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
32	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***2577K*** Click on **Initialize Plan Title: Test of OPLAN *2577K*** Click on **Plan Type: Normal** Click on **Plan Distribution: Local** Click on **Database Area: Real World** Click on **Plan Classification: Unclassified**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 2577K.		
33	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
34	Using SQL*Plus, run the INITHT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the INITDS10.001 data sheet.		
35	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***3577K*** Click on **Initialize Plan Title: Test of OPLAN 3577K** Click on **Plan Type: Restricted** Click on **Enter Userids: USER IDS: *SMTEST2* Transmit** Click on **Plan Distribution: Local** Click on **Database Area: Real World** Click on **Plan Classification: Unclassified**	A INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 3577K.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
36	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A INITHT transaction should be present on the receive queues of both servers.		
37	Using SQL*Plus, run the INITHT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the INITDS10.002 data sheet.		
38	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***3577K*** Click on **Modify** Click on **Enter Userids USER IDS: *SMTEST3***	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add user SMTEST3 to the restricted users list for OPLAN 3577K.		
39	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
40	Using SQL*Plus, run the INITHT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the INITDS10.004 data sheet.		
41	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***3577K*** Click on **Modify** Click on **Enter Userids** Click on **Delete** (for SMTEST2) Click on **Delete** (for SMTEST3)	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete users SMTEST2 and SMTEST3 from the restricted user list from OPLAN 3577K.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
42	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
43	Using SQL*Plus, run the INITHT.SQL script on the Source, Destination, and Distant servers, and start up the System Services application server	The expected results should match those found in the INITDS10.005 data sheet.		
44	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***3577K*** Click on **Modify** Click on **Plan Type: Normal**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will change OPLAN 3577K from being restricted to normal.		
45	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
46	Using SQL*Plus, run the INITHT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the INITDS10.006 data sheet.		
47	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***2577K*** Click on **Delete**	A RESTHT and two SYNCHT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will delete OPLAN 2577K.		

TEST CASE 41422-10: TRANS-CH-PLAN-INIT (INITHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
48	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***3577K*** Click on **Delete**	A RESTHT and two SYNCHT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. These transactions will delete OPLAN 3577K.		
49	Verify that all of the INITHT transactions generated were received at the temporary directory created for XTDS for all three servers	The INITHT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
50	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)	
Purpose: The purpose of this test is to verify that the software correctly processes the delete OPLAN functions of the TRANS-IRM-DATA (RESTHT) transaction.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 3 transaction flat files used with XTP c. 2 SQL script files d. 3 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and IRM sources.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Services applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	If OPLAN 2577K already exists in the previous step, then skip the following two steps If OPLAN 2577K does not exist, the next two steps will create it	OPLAN 2577K is required to continue this test case.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff11.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create and initialize OPLAN 2577K for RESTHT and CDAYHT testing.		
8	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
9	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INITDS11.001 data sheets.		
10	From the SQL> prompt on all three servers, type ***start cdayht*** to run the CDAYHT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
11	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/cdayff11.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A CDAYHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will set the CDAY to Oct 10, 1995 and the LHOUR to 10:10 AM for OPLAN 2577K.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A CDAYHT transaction should be present on the receive queues of both servers.		
13	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start cdayht*** to run the CDAYHT.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the CDAYDS11.001 data sheets.		
14	From the SQL> prompt on all three servers, type ***start restht*** to run the RESTHT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
15	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/restff11.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will reset the CDAY and LHOUR to blanks.		
16	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A RESTHT transaction should be present on the receive queues of both servers.		
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start restht*** to run the RESTHT.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the RESTDS11.001 data sheets.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	From the SQL> prompt on all three servers, type ***start restht2*** to run the RESTHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
19	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/restff11.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will set the status code to 4, which prepares OPLAN 2577K for deletion.		
20	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A RESTHT transaction should be present on the receive queues of both servers.		
21	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start restht2*** to run the RESTHT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the RESTDS11.002 data sheets.		
22	From the SQL> prompt on all three servers, type ***start syncht1*** to run the SYNCHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
23	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/syncff11.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2577K.		
24	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SYNCHT transaction should be present on the receive queues of both servers.		
25	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start syncht1*** to run the SYNCHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SYNCDS11.001 data sheets. This will exit SQL*Plus and start up System Services.		
26	Verify that all of the previous RESTHT and CDAYHT transactions generated were received at the temporary directory created for XTDS for all three servers	The RESTHT and CDAYHT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
IRM Transaction Testing				
27	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Services applications are successful.		
28	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
29	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
30	Enter and transmit the following data: OPLAN: ***2577K*** Click on **Initialize PLAN TITLE: *TEST 11*** Click on **Plan Type: Normal** Click on **Plan Distribution: Local** Click on **Database Area: Real World** Click on **Plan Classification: Unclassified**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 2577K for the purposes of testing RESTHT and CDAYHT.		
31	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
32	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***2577K*** Click on **Modify** Click on **TCC Indicators** Click on **Reset AMC Schedules from *C002* to *C014***	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will reset AMC Schedules for OPLAN 2577K.		
33	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A RESTHT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-11: TRANS-IRM-DATA (CDAYHT/RESTHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
34	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***2577K*** Click on **Modify** CDAY ***01Oct95*** LHOUR ***0800***	A CDAYHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will set the C-DAY of OPLAN 2577K for October 1, 1995 at 8:00 A.M.		
35	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A CDAYHT transaction should be present on the receive queues of both servers.		
36	Using IRM on the Source server, enter and transmit the following data: OPLAN: ***2577K*** Click on **Delete**	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2577K.		
37	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A RESTHT transaction should be present on the receive queues of both servers.		
38	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)	
Purpose: The purpose of this test is to verify that the software correctly processes TRANS-PLAN-DATA (PLNUAT) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 7 transaction flat files used with XTP c. 1 SQL script files d. 7 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and IRM sources. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat1*** to run the PLNUAT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Major Forces for OPLAN 2591H.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PLNUAT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat1*** to run the PLNUAT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.001 data sheets.		
9	From the SQL> prompt on all three servers, type ***start plnuat2*** to run the PLNUAT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
10	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Constraints, OPLAN Critical Resources, and the OPLAN Supply Shortfall for OPLAN 2591H.		
11	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PLNUAT transaction should be present on the receive queues of both servers.		
12	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat2*** to run the PLNUAT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.002 data sheets.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
13	From the SQL> prompt on all three servers, type ***start plnuat3*** to run the PLNUAT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
14	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Concept and Supporting OPLAN for OPLAN 2591H.		
15	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PLNUAT transactions should be present on the receive queues of both servers.		
16	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat3*** to run the PLNUAT3.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.003 data sheets.		
17	From the SQL> prompt on all three servers, type ***start plnuat4*** to run the PLNUAT4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Narrative and Key Assumptions for OPLAN 2591H.		
19	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	All PLNUAT transactions should be present on the receive queues of both servers.		
20	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat4*** to run the PLNUAT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.004 data sheets.		
21	From the SQL> prompt on all three servers, type ***start plnuat5*** to run the PLNUAT5.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
22	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Mission, OPLAN Condition, and Objective Area for OPLAN 2591H.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
23	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	All PLNUAT transactions should be present on the receive queues of both servers.		
24	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat5*** to run the PLNUAT5.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.005 data sheets.		
25	From the SQL> prompt on all three servers, type ***start plnuat6*** to run the PLNUAT6.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
26	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will update the OPLAN Resupply Shortfall and Related Personnel Shortfall for OPLAN 2591H.		
27	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	All PLNUAT transactions should be present on the receive queues of both servers.		
28	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat6*** to run the PLNUAT6.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.006 data sheets.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
29	From the SQL> prompt on all three servers, type ***start plnuat7*** to run the PLNUAT7.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
30	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/plnuff12.007*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete all of the previously updated sections for OPLAN 2591H.		
31	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	All PLNUAT transactions should be present on the receive queues of both servers.		
32	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start plnuat7*** to run the PLNUAT7.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the PLNUDS12.007 data sheets. This will exit SQL*Plus and start up System Services.		
IRM Transaction Testing				
33	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
34	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
35	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
36	Enter and transmit the following data: OPLAN: ***1106K*** Click on **Initialize PLAN TITLE: *1106K TEST*** Click on **Plan Type: Normal** Click on **Plan Distribution: Local** Click on **Database Area: Real World** Click on **Plan Classification: Unclassified**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 1106K.		
37	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
38	Using SQL*Plus, run the INITHT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	The expected results should match those found in the INITDS12.001 data sheet.		
39	From the Plan Maintenance screen, enter and transmit the following data: OPLAN: ***1106K*** Click on **Modify** Click on **Lock**	A PLNUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will change the status of OPLAN 1106K from Unlocked to Locked.		

TEST CASE 41422-12: TRANS-PLAN-DATA (PLNUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
40	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PLNUAT transaction should be present on the receive queues of both servers.		
41	Using SQL*Plus, run the PLNUAT.SQL scripts on the Source, Destination, and Distant servers, and start up the Systems Services application server	OPLAN 1106K will have a change in status. The expected results should match those found in the PLNUDS12.007 data sheet.		
42	From the Plan Maintenance screen, enter and transmit the following data: OPLAN: ***1106K*** Click on **Delete**	A RESTHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 1106K.		
43	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A RESTHT transaction should be present on the receive queues of both servers.		
44	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)	
Purpose: The purpose of this test is to verify that the software correctly processes the TRANS-SYNC-DATA (SYNCHT) transaction which is used to request update of synchronization data for an OPLAN.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 5 transaction flat files used with XTP c. 2 SQL script files d. 3 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP and IRM sources.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login and bring up System Services on all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Ensure that a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	If OPLAN 2577K already exists in the previous step, then skip the following two steps If OPLAN 2577K does not exist, the next two steps will create it	OPLAN 2577K is required to continue this test case.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff13.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create and initialize OPLAN 2577K (local) for SYNCHT testing.		
8	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
9	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht1*** to run the INITHT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the INITDS13.001 data sheets.		
10	From the SQL> prompt on all three servers, type ***start syncht1*** to run the SYNCHT1.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
11	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/syncff13.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SYNCHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will set the OPLAN status code to a '*' (from 0), which sets up OPLAN 2577K for deletion.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
12	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SYNCHT transaction should be present on the receive queues of both servers.		
13	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start syncht1*** to run the SYNCHT1.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SYNCDS13.001 data sheets.		
14	From the SQL> prompt on all three servers, type ***start syncht2*** to run the SYNCHT2.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
15	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/syncff13.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SYNCHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2577K.		
16	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SYNCHT transaction should be present on the receive queues of both servers.		
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start syncht2*** to run the SYNCHT2.SQL script	This will begin SQL*Plus. The data in the database should match the data found in the SYNCDS13.002 data sheets.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	From the SQL> prompt on all three servers, type ***start initht4*** to run the INITHT4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This is a preliminary check of the database to see what data exists before the next transaction is processed. This will exit SQL*Plus and start up System Services.		
19	If OPLAN 2566K already exists in the previous step, then skip the following two steps If OPLAN 2566K does not exist, the next two steps will create it	OPLAN 2566K is required to continue this test case.		
20	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/initff13.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create and initialize OPLAN 2566K for SYNCHT testing.		
21	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
22	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start initht4*** to run the INITHT4.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the INITDS13.002 data sheets. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
24	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
25	Enter and transmit the following data: OPLAN: ***2566K*** Click on **Modify** Click on **Plan Distribution: Distributed** Click on **Enter Sites** SITE IDs: ***(a valid site)***	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will make OPLAN 2566K a distributed OPLAN.		
26	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start syncht3*** to run the SYNCHT3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
27	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
28	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/counts13.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SYNCHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will populate the counts for AMC, MTMC., MSC, Organic Carriers, and ULNs for OPLAN 2566K.		
29	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A set of ULNUBT and NRNUBT transactions should be present on the receive queues of both servers.		
30	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/syncff13.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A SYNCHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will tally the counts for AMC, MTMC., MSC, Organic Carriers, and ULNs for OPLAN 2566K.		
31	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SYNCHT transaction should be present on the receive queues of both servers.		
32	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start syncht3*** to run the SYNCHT3.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the SYNCDS13.003 data sheets. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
33	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
34	Enter and transmit the following data: OPLAN: ***2566K*** Click on **Delete**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2566K.		
35	Verify that all of the previous SYNCHT transactions generated were received at the temporary directory created for XTDS for all three servers	The SYNCHT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		
IRM Transaction Testing				
36	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
37	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
38	On the Source server, from the **System Services** menu (SS-000-1), click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
39	Enter and transmit the following data: OPLAN: ***2577K*** Click on **Initialize** TITLE: ***Title for 2577K*** Click on **Plan Type: Normal** Click on **Plan Distribution: Local** Click on **Database Area: Real World** Click on **Plan Classification: Unclassified**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 2577K.		
40	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
41	Under the Plan Maintenance menu, enter and transmit the following data: OPLAN: ***2577K*** Click on **Delete**	Two SYNCHT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2577K.		
42	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	Two SYNCHT transactions should be present on the receive queues of both servers.		
43	On the Source server, from the **System Services** menu (SS-000-1), click on **GCCS System Services ▸ Plan Management ▸ Plan Network Status**	The **Plan Network Status** screen appears (SS-IRM-13).		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
44	Enter and transmit the following data: OPLAN: ***2566K*** Click on **Initialize** TITLE: ***Title for 2566K*** Click on **Plan Type: Normal** Click on **Plan Distribution: Distributed** Click on **Enter Sites** SITE IDs: ***(a valid site)*** click on **[TRANSMIT]** twice Click on **Database Area: Real World** Click on **Plan Classification: Unclassified** click on **[TRANSMIT]**	An INITHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will create OPLAN 2566K.		
45	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	An INITHT transaction should be present on the receive queues of both servers.		
46	Using SQL*Plus, run the SYNCHT.SQL scripts on the Source server	The expected results should match those found in the SYNCDS13.002 data sheet.		
47	On the Source server, from the **System Services** menu (SS-000-1), click on **GCCS System Services ▸ Plan Management ▸ Plan Network Status**	The **Plan Network Status** screen appears (SS-IRM-13).		
48	Enter and transmit the following: OPLAN: ***2566K*** Click on **Generate New SYNC Requests**	A SYNCHT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will tally the counts of requirements for OPLAN 2566K on the sites indicated.		

TEST CASE 41422-13: TRANS-SYNC-DATA (SYNCHT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
49	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A SYNCHT transaction should be present on the receive queues of both servers.		
50	Using SQL*Plus, run the SYNCHT.SQL scripts on the Source server	The expected results should match those found in the SYNCDS13.003 data sheet.		
51	On the Source server, from the **System Services** menu (SS-000-1), click on **GCCS System Services ▸ Plan Management ▸ Plan Maintenance**	The **Plan Maintenance** screen appears (SS-IRM-1).		
52	Enter and transmit the following data: OPLAN: ***2566K*** Click on **Delete**	Two SYNCHT transactions will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete OPLAN 2566K.		
53	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	Two SYNCHT transactions should be present on the receive queues of both servers.		
54	Using SQL*Plus, run the SYNCHT.SQL scripts on the Source server	The expected results should match those found in the SYNCDS13.002 data sheet.		
55	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)	
Purpose: The purpose of this test is to verify that the software correctly processes TRANS-AGENCY-DATA (PFMUAT) transactions.	
Tester Information: Tester Name: Phone Number: Date(s) of Test:	**Prerequisites for this test:** a. 3 servers with the proper permissions established b. 8 transaction flat files used with XTP c. 1 SQL script files d. 8 result data sheets e. OPLAN 2591H exists and is a copy of 2500T
Test Connectivity: Server ID/Terminal ID: Source server is JDIC2 Server ID/Terminal ID: Destination server is JDIC3 Server ID/Terminal ID: Distant server is JDIC4	**Software Version:** a. b. c.

NOTES:

1. Distribution of transactions designated as network will be generated at the Source server and distributed to the Destination and Distant servers by way of Transaction Distribution Services.

2. This transaction will be tested from the XTP. This transaction had been modified thus the new functionality will be specifically exercised. This test will verify that the new functionality defined in the informal spec (RTS) functions properly.

3. This test will be conducted with three servers: a Source server, a Destination server, and a Distant server. During transaction testing, the testers will attempt to use the Destination and Distant servers an equal amount of time.

4. Transaction files with the extension .SQL that are used in the test case are located in the /h/SM/data/tds/krh/k_sql directory.

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
1	Login to all three servers: Source, Destination, and Distant servers, and start up the System Services application	All logins and System Service applications are successful.		
2	Turn Journalling on under System Services, clear out the receive queues for each server, and turn Journalling off	The receive queue will be empty on each server.		
3	Add a temporary directory exists in the Subscriber file for the purpose of testing XTDS on all three servers	A temporary directory will exist on each server (i.e., xtds_dir).		
4	On the Source server, from the **System Services** menu (SS-000-1), go to the following: Click on **GCCS System Services ▸ System Services Utilities ▸ External Transaction Processor**	The **External Transaction Processor** screen appears (SS-UTL-4).		
5	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start pfmuat*** to run the PFMUAT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. This is a preliminary check of the database to see what data exists before any transactions are processed. This will exit SQL*Plus and start up System Services.		
6	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.001*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will add ULNs PAA and PAB, CINs 20004 and 20009, and PINs 21002 and 21003 for specified providing organizations corresponding to OPLAN 2591H, performing an update.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
7	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PFMUAT transaction should be present on the receive queues of both servers.		
8	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start pfmuat*** to run the PFMUAT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the PFMUDS14.001 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
9	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.002*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will test the Trans_Frns_In_Use field, now an obsolete field that was once used with the main frame. If the field contains data, then processing the transaction should not affect the database in any way. No error message will appear.		
10	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PFMUAT transaction should be present on the receive queues of both servers.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
11	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start pfmuat*** to run the PFMUAT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the PFMUDS14.002 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		
12	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.003*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will test the Trans_Agency_In_Use field, a second obsolete field that was once used with the main frame. If the field contains data, then processing the transaction should not affect the database. No error message will appear.		
13	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PFMUAT transaction should be present on the receive queues of both servers.		
14	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start pfmuat*** to run the PFMUAT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the PFMUDS14.003 data sheet. Remember this data as a check for the next transaction. This will exit SQL*Plus and start up System Services.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
15	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.004*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will be generated on the Source server and forwarded to the Destination and Distant servers' receive queues. This transaction will delete ULNs PAA and PAB, CINs 20004 and 20009, and PINs 21002 and 21003 from OPLAN 2591H.		
16	Verify that the transaction was received and logged on the Destination and Distant servers using the **GCCS System Services ▸ Monitors ▸ Receive Queue Detail** screen	A PFMUAT transaction should be present on the receive queues of both servers.		
17	From the UNIX prompt on all three servers, type ***sqlplus /*** From the SQL> prompt on all three servers, type ***start pfmuat*** to run the PFMUAT.SQL script Type ***quit*** to exit SQL and ***start_ss*** from the UNIX prompt	This will begin SQL*Plus. The data in the database should match the data found in the PFMUDS14.004 data sheet. This will exit SQL*Plus and start up System Services.		
18	Verify that all of the previous PFMUAT transactions generated were received at the temporary directory created for XTDS for all three servers	The PFMUAT transactions should be found in the files listed in the temporary directory set up for XTDS. The transaction naming convention of the files can be found in Appendix A of the SMM.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
Invalid Field Testing				
19	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.005*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will not be generated. This transaction will test for a null in the OPLAN field. An error will appear and be recorded in the **msg** file stating that the OPLAN does not exist in the database.		
20	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.006*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will not be generated. This transaction will test for an invalid Providing Organization Code. An error message will appear and be recorded in the **msg** file.		
	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.007*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will not be generated. This transaction will test for an invalid Total FRN (non-numeric). An error message will appear and be recorded in the **msg** file.		

TEST CASE 41422-14: TRANS-AGENCY-DATA (PFMUAT)				
STEP	**TEST STEP / INPUT**	**EXPECTED RESULTS / COMMENTS**	**PASS**	**FAIL/PR**
18	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.008*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will not be generated. This transaction will test for an invalid First CIN (non-numeric). An error message will appear and be recorded in the **msg** file.		
19	Enter the following: Transaction File Name: ***/h/SM/data/tds/krh/pfmuff14.009*** Message Log File Name: ***/export/home/msg*** Report File Name: ***/export/home/rpt*** Click on **Update Database With Valid Transactions Transmit**	A PFMUAT transaction will not be generated. This transaction will test for an invalid Last PIN (non-numeric). An error message will appear and be recorded in the **msg** file.		
20	Using the Audit Reports, obtain a hard copy print out of all the transactions generated during the previous steps Run an audit report for each server used during this test	Review of transactions indicates proper transaction processing on each server. Also each server shows each of the transactions generated at the Source server.		

TEST CASE 41422-15: TRANS-USERID-AC (USERHT)	
Purpose:	
Tester Information:	**Prerequisites for this test:**
Test Connectivity:	**Software Version:**
NOTES: **NOT APPLICABLE -- REQUIREMENT CHANGE**	

THIS TEST CASE PAGE INTENTIONALLY LEFT BLANK.